2019HW行动防守总结

原创： 瓦都剋

> 从五月19日到六月底一直在参与HW行动，第一次参与从甲方角度的攻防对抗的团队当中，总体来说感触很大，谨此记录下。

0x00 前言

首先强调下，下文所有的思路均是"因地制宜"，根据客户业务实际情况开展。这次HW是跟某国企合作一块进行安全防御，之前一直是做的乙方的安全服务，第一次切换到防护角色。

首先，整个项目分为两个阶段：

1. 护网前
2. 护网中

0x01 护网前

护网前的工作相当重要，国企的安全工作想必各位都有所了解这里就不多说了，如何在这么多的时间内尽量做到全面安全覆盖实在是太考验防守团队了。

前期的准备工作主要分为两个部分：

1、自检

2、加固

自检是最快发现问题的手段，本次项目我们负责的是云平台安全，云主机总数量达14000多台，任务还是蛮重的。

1.1 自检

主要分为外网渗透测试和内网渗透测试两部分，外网和内网渗透测试还不一样，外网需要结合客户相关业务来，比如政企客户你使用WordPress、Drupal的payload去打，不能说百分之百没有，但是成功的概率相当低，要有针对性去测试，因为时间不允许去这么做。

1.1.1 外网渗透测试

外网不是我们的重点，根据实际情况主要关注那些能直接获取主机权限的漏洞和泄漏大量数据的漏洞即可，比如：

弱口令：数据库、SSH、RDP、后台

命令执行：Solr、Jenkins、Weblogic、Struts2、RMI、JBoss、Tomcat、Spring、ActiveMQ、Zabbix

文件操作：文件上传、文件读取

未授权访问：Redis、Hadoop、Docker、K8S

1.1.2 内网渗透测试

内网渗透和外网渗透区别很大，因为外网不仅有我们，还有其他合作伙伴，各种WAF、防火墙、APT、IDS、IPS、以及办公网的EDR、杀毒什么的各种安全设备一顿操作，所以我们此次的重心就是在内网云平台中，我们的靶标当然也在内网云平台中。

由于内网云平台有很多微服务集群，所以根据这个特性，主要分以下几个步骤：

1、资产发现

因为云平台都是在专有云上，所以写个脚本爬了下资产就可以全部获取到了。

这里需要注意下爬的字段，因为后期涉及到漏洞修复、安全事件等问题的对接。

| IP | 项目 | 部门 | 状态 |

2、资产指纹梳理

因为主机数量太大，如果每个资产单独去测试的话，时间代价太大，所以很有必要进行资产指纹梳理，这样如果发现某个漏洞，直接就可以为后面的批量漏洞利用提供条件了。

资产指纹根据业务情况进行有针对性的收集，主要搜集的有：

21、22、23、1090、1099、2049、2181、3000、3306、4848、5000、5900、5901、6379、7000-9999、11211、12181、27017、10050、50000、50080

3、漏洞发现

漏洞发现就是常规的渗透测试了，但是内网的渗透有别与外网，内网主要是快速的，尽可能的发现更多的漏洞，所以不需要进行进一步利用，比如发现了Redis未授权访问，就不要去浪费时间反弹shell、写密钥这些。

这里举了个两个简单的例子

Example1

https://192.168.2.11:8888/PS：前期资产发现阶段发现8888端口对应的主机有1800多台1、直接访问提示4032、Fuzz目录发现存在taskFile，https://192.168.2.11:8888/taskFile3、访问https://192.168.2.11:8888/taskFile提示参数不对4、Fuzz参数发现参数tId，https://192.168.2.11:8888/taskFile?tId=5、输入值，访问https://192.168.2.11:8888/taskFile?tId=1发现提示tId-1不存在6、删除值访问https://192.168.2.11:8888/taskFile?tId=发现提示/root/xxx/xxx任务不存在7、Fuzz漏洞，成功发现本地文件包含https://192.168.2.11:8888/taskFile?tId=../../../../etc/passwd

Example2

https://192.168.2.11:8006/PS：前期资产发现阶段发现8006端口对应的主机有180多台1、直接访问https://192.168.2.11:8006/env，发现存在SpringBoot Actuator信息泄漏/autoconfig/beans/env/configprop/dump/health/info/mappings/metrics/shutdown/trace/env2、访问https://192.168.2.11:8006/trace，发现存在访问URL：/application.wadl3、直接访问发现信息泄漏，可直接看到整站的目录结构/war/war/path...4、一层一层访问目录接口发现/war/WEB-INF/classes目录下存在redis.properties文件，直接访问提示4035、根据前期漏洞信息、收集信息关联分析，发现在对应文件下加入content即可进行文件读取：/war/WEB-INF/classes/redis.properties/content这样就可以直接导致web目录下任意文件读取了，批量利用发发现8006端口全部主机均存在此漏洞6、访问https://192.168.2.11:8006/war/WEB-INF/classes/redis.properties/content读取到redis密码，为弱口令：Nb1234567、根据整理的资产指纹信息编写批量利用脚本，发现60多台Redis存在此弱口令

4、批量漏洞利用

这部分就是批量发现涉及的问题主机了，大部分是需要写脚本的，如果有相关工具那最好。

比如，我们在上面的发现的漏洞，利用收集的资产指纹信息直接编写脚本批量跑一下就行了

当然其他漏洞也一样，这里我主要以我们测试的几个典型批量为例：

SSH、FTP、MySQL、Zoomkeeper、MongoDB、Hadoop、Redis、Struts2、Weblogic、Docker、OpenSSL、Werkzeug、Jboss、ActiveMQ、Zabbix、K8S、Druid等。

├── IP.txt ├── Zookeeper │ ├── vul.txt │ └── zookeeper_unauth_access.py ├── bash ├── ftp ├── memcache ├── mongodb ├── mssql ├── openssl ├── postgresql ├── redis ├── smb ├── ssh ├── st2 ├── weblogic ├── druid ├── zabbix ├── docker... └── all_poc.py

这里是我们之前做基线检查的相关脚本，然后根据需求改了下，也可以用POC-T、Pocsuite等这些框架，反正能用就行了。

整个内网渗透下来战果还是很大的，拿下了我们护航的两个靶标系统、两个统一运维平台、数据管理平台、热更新管理平台、多个Master、大量SSH等弱口令。

这就为后面安全加固打了个好基础。

1.2 加固

1.2.1 运维相关

1、测试、开发服务器全部关停

2、敏感端口安全组隔离

3、使用堡垒机、跳板机运维

4、用户、权限隔离

这里特别说明下，比如：MySQL不使用默认root用户，使用业务/项目名中间用下划线连接的方式，因为爆破都是默认用户的，这样即使存在弱口令，也将安全性提升一大截

5、认证策略：密钥、双因子认证

6、最小化权限原则

7、补丁

1.2.2 安全设备

这里我们只说下使用了我们云平台的相关产品：

WAF：使用防火墙，防御大多数攻击。

流量、日志、数据库审计：流量、日志、数据库方便万一出现安全事故，能够快速响应，溯源分析找到问题源头

态势感知：这个就不说了。

安骑士：主机安全，能够第一时间发现异常登录、异常进程、异常网络连接、后门、账户等高危安全问题，并且能快速定位问题所在，相当有用。

蜜罐：攻击者进入内网后会进行内网渗透，收集信息，攻击靶标，所以部署蜜罐是发现内网异常的必要手段。主要部署了常用的服务，比如SSH、MySQL、FTP、MongoDB、Redis、Weblogic、Struts2、Tomcat、Joomla、PostgreSQL、Shellshock、SMB、Memcache、Telnet。

扫描器：由于主机量巨大，手工总会有已漏，所以在不同网段部署扫描器，每天在0点到1点自动开始巡检。

1.2.3 环境隔离

防火墙策略：虽然安全组也可以做，但是内网业务层面还需要在防火墙上做策略。

堡垒机：使用堡垒机是为了保证运维安全，同时保障重要系统及靶标隔离。堡垒机的登录也是有访问控制的，只允许白名单IP访问，同时堡垒机使用强口令+随机KEY登录。

双因子认证：重要系统(控制台，总控系统，运维系统等)有必要的均采用堡垒机登录，无法使用的采用白名单IP+强口令策略+随机验证码组合策略。

靶标：靶标乃重中之重，一旦被攻破，直接可以收拾收拾回家了，脸面不说，重要的是钱可能都没了。靶标是只允许堡垒机IP能够访问，同时我们前期花了很大功夫渗透了一波，删除不必要账户，所有账户使用十六位随机密码+随机6位KEY登录。

0x02 护网中

2.1 查漏补缺

1、虽然前期做了渗透测试，但毕竟也就一周时间，所以在护网期间我们在云管控平台上找了下端口对应主机数超过20个的服务，然后也成功发现了两三个高危漏洞，影响一千多台主机。

2、数据库查看端口对应主机的时候发现了一些非标准端口，比如MySQL数据库的3303、3304、3305、3307，Redis的6377、6378等，又成功破解了几十台弱口令。

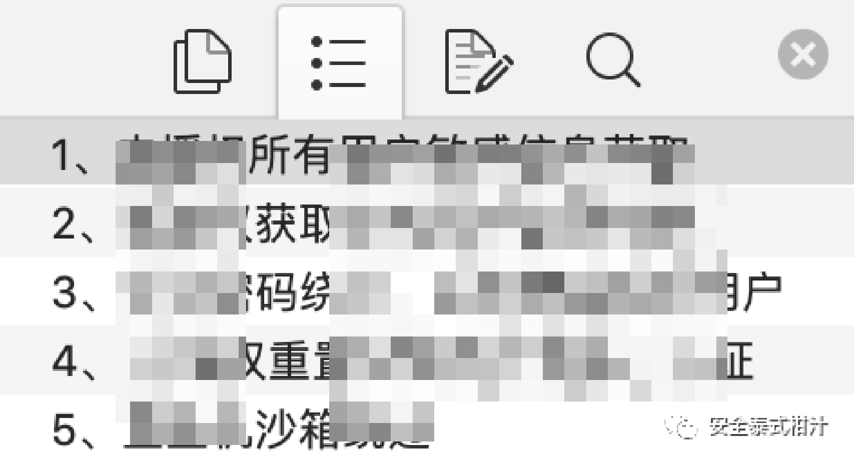
3、通过日志检索、流量分析等发现了一些弱口令，利用这些弱口令批量测试了一遍云平台所有主机，发现大量SSH、MySQL、Redis弱口令。

4、在进行全量日志检索的时候，发现一些运维直接将一些敏感字段直接写在了命令行中，比如MySQL连接是直接将密码写在了命令行，这样如果一台主机被入侵，那么history是必看的，如果该密码是通用的将直接或间接的影响更多的主机，所以统一安排将history命令历史清空，再统一改了下相关配置。

5、新建主机风险，在查看主机对应业务的时候发现了在护网开始后还是会有新主机的创建，那么这些新建的主机就存在较大的安全风险了。

6、堡垒机安全问题，堡垒机我们是采用白名单IP+强密码+令牌的登录方式，但是护网期间随意看下了，还搞了该厂商堡垒机的好几个0day。前台某接口可直接获取全部用户包括明文密码等全部敏感信

息、重置任意用户密码、更换认证方式及客户端限制等。此外，堡垒机登录后还可以绕过沙箱限制，做一些限制的操作。



公网上找了下这个堡垒机，直接默认密码就分分钟控制几百台肉🐔。。。

堪忧啊~

2.2 安全告警

主要就是安全设备上的安全告警了，比如蜜罐的端口探测告警、蜜罐入侵事件；态势感知的暴力破解事件、攻击事件；安骑士的后门、网络连接、进程、账户等；WAF封IP一把梭。

中间出现个插曲，扫描器和蜜罐都出了问题，扫描器逻辑bug就不说了，主要是蜜罐。安骑士告警蜜罐控制台IP存在异常命令执行，告警信息是由weblogic调用了bash并执行了`cat /etc/passwd`，经过一上午分析，发现是部署了weblogic蜜罐的宿主机上会有和weblogic蜜罐一模一样的进程，又由于蜜罐出口都是经过宿主机，所以宿主机上的安骑士产生了告警。

事件是解决了，但是蜜罐中的进程为什么会在宿主机中存在一个clone版的，这是什么操作？？？

另外，万一蜜罐使用的Docker版本存在逃逸，在没有主机安全产品保护的情况下，那岂不是一个活靶子？

2.3 事件分析

这就是应急响应、事件分析与溯源的范畴了。

说个小事，其他厂商告警存在恶意文件下载，地址是我们OSS的一个地址，OSS是一个静态存储服务器，供业务应用上传媒体文件、静态资源文件等使用，一个XSS了不起了，能存在什么漏洞？领导搞的相当慌，五分钟内让删除，这个安全事件很简单：

1、如果访问的IP是业务服务器

- 要么AK泄漏了直接利用AK进行的文件增删改查；
- 要么业务上传的地方没有做严格限制，导致了任意格式文件上传；
- 该服务器应该还存在SSRF漏洞；

2、如果访问的IP不是业务服务器

由于OSS上传后会生成随机字符串的URL，根本是无法猜到的，那么谁访问了这个URL，那么就是这个IP上传的，这个IP就是有问题的。

2.4 防火墙策略

因为我们是负责云平台防护，所以我们主要是找一些和云平台有直接交互的业务，然后就在防火墙上做了下策略。这些我就不清楚了，专业的人干专业的事嘛。

2.5 WAF等

不管三七二十一、先封为敬，另外一定要及时关注安全预警，直接在WAF上做自定义规则，比如weblogic的，直接拦截`_async`，`bea_wls_internal`等URL路径即可。

2.6 安全组

发现敏感端口、高危漏洞，先把安全组做了。

2.7 护网神操作

护网期间听闻的各种神操作：

- 无人机搞Wi-Fi的
- 挖下水道进内部的
- BadUSB钓鱼的
- 提前半个月混入甲方的
- 望远镜看密码的
- 红队伪装蓝队进行木马钓鱼的
- 利用深信服0day伪装进入的
- 首页开局一张图片的
- 美人计的

- ...

感谢各位天秀大佬们的免费教程！学习了！

2.8 问题思考

1、蜜罐的局限性

我所了解的市面上的蜜罐和业务基本上都是无关联、无数据的，对于攻击队那些资深大佬来说，我个人觉得应该采用的是"深度优先原则"的方式进行攻击，因为这种方式内网动作更小，更不易察觉。

基于这个局限性，蜜罐就很容易被识破了，比如查看hosts文件、history、网段、数据、业务等。

2、安全产品

堡垒机、VPN、蜜罐、扫描器等这些安全产品，比如说扫描器，基本上告警源是扫描器的，都肯定是忽略了，那万一扫描器被人家搞了呢？对于安全公司的相关产品，天天对外向企业吧啦吧啦什么SDLC，自己的产品上线前都不做安全测试的嘛？？？

0x03 感触

1、整个护网三周，除了一些小打小闹，基本上没有看到真实的攻击，搞的很慌。。。真怕最后一天被淘汰了。。。

2、根据"HW-2019第一天蓝方总结，险遭RDP0 day打击"这个钓鱼文章，联想到，作为防守方也可以钓红队呀。

举个例子：在Github上伪造个含公司信息的代码仓库，故意放些密码，比如MySQL密码，然后攻击队肯定会去GitHub搜索敏感信息，然后就会尝试利用。。。@#$%^&*

后续我们又一块头脑风暴了下，整理了好多搞红队的途径，既精准又快，期待下次试试水。

3、云真的方便，各种好用。

4、平台真的很重要！

阅读原文